



Deliverable-01

WIS Architecture Report

Workgroup: E7.04

Date: 02/03/2022

Miembros:

Baños González, Alejandro (alebangon@alum.us.es)

Flores Rodríguez, Román (romflorod@alum.us.es)

Grosso Gómez de Terreros, Javier (javgrogom@alum.us.es)

Gutiérrez Ceballos, Pablo (pabgutceb@alum.us.es)

Ibáñez Montero, Julia (julibamon@alum.us.es)

Roldán Cadena, Jesús (jesroldcad@alum.us.es)

Repositorio de Github: <https://github.com/jesroldcad/Acme-Toolkits>

Índice

1. Resumen ejecutivo	1
2. Tabla de control de cambios	2
3. Introducción	3
4. Arquitecturas de Sistemas de Información Web	4
5. Experiencia previa del equipo	4
5.1. Arquitectura REST	4
5.2. Patrón MVC	5
6. Conclusiones	5
7. Referencias bibliográficas	6

1. Resumen ejecutivo

Este documento detalla los conocimientos básicos con los que contamos en relación a la arquitectura y los patrones de diseño software, así como una breve descripción de los mismos.

Aunque hemos estudiado múltiples tipos de arquitecturas y patrones, hablaremos de los que conocemos con mayor profundidad o con los que el equipo tiene más experiencia general (usados en otros proyectos anteriormente).

Ya que hemos puesto en común la experiencia de todos los miembros del equipo, hemos concluido que nuestro conocimiento en las arquitecturas software parte de la misma base y que, por tanto, manejamos (sobre todo) las mismas arquitecturas y patrones de sistemas de información web. Por tanto, se describen a nivel general los servicios de arquitectura Cliente-Servidor REST y el patrón arquitectónico Modelo Vista Controlador (MVC), ya que los hemos estudiado y utilizado para el desarrollo de distintas aplicaciones software a lo largo de nuestra vida y tienen la mayor parte del peso en nuestra base de conocimiento de arquitecturas software.

En conclusión, creemos que, aunque nos queda mucho por aprender, tenemos una base general suficiente para poder enfrentarnos al nuevo proyecto que nos brinda la asignatura Diseño y Pruebas II.

2. Tabla de control de cambios

Número de revisión	Descripción	Fecha
1	Creación del documento acorde a la plantilla.	24/02/2022
2	Completados todos los apartados del documento.	25/02/2022
3	Revisión general del documento. Se han realizado pequeñas correcciones de forma y contenido.	26/02/2022

3. Introducción

En este documento se va a reflejar la experiencia y conocimiento que tiene el equipo en relación con las arquitecturas de Sistemas de Información Web, así como una breve descripción de las mismas. Para ello, hemos organizado este documento en distintas secciones descritas a continuación.

Sección 4. *Arquitecturas de Sistemas de Información Web.* Antes de hablar de las arquitecturas con las que estamos más familiarizados, se hará una breve descripción de qué son estas arquitecturas software, así como qué es un patrón arquitectónico y qué ventajas aporta.

Sección 5. *Experiencia previa del equipo (5.1 Patrón MVC, 5.2 Arquitectura REST).* En esta sección se hará una breve descripción de la base de conocimiento desde la que parte el equipo en cuanto a arquitecturas y patrones de sistemas web. En cada uno de los subapartados se van a describir de manera breve cada una de ellas.

Sección 6. *Conclusiones.* Se reflejarán las conclusiones que hemos compartido todo el equipo acerca de los conocimientos previos que tenemos en común y cómo pensamos que nos vamos a enfrentar a la asignatura Diseño y Pruebas II.

Sección 7. *Referencias bibliográficas.* En esta sección reflejaremos las fuentes que hemos utilizado para recopilar alguna información necesaria para el correcto desarrollo de este informe.

Finalmente, el documento posee la siguiente estructura: portada, índice, resumen ejecutivo, tabla de control de cambios, introducción, contenidos (divididos en las secciones *Arquitecturas de Sistemas de Información Web* y *Experiencia previa del equipo*, dividida a su vez en *Patrón MVC* y *Arquitectura REST*).

4. Arquitecturas de Sistemas de Información Web

Una arquitectura de software, también denominada arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software. Define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos.

Los patrones arquitectónicos, también denominados patrones de arquitectura o arquetipos, ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados.

5. Experiencia previa del equipo

Todos los miembros del equipo hemos estudiado qué son las arquitecturas software, así como los patrones. Teóricamente conocemos múltiples ejemplos de arquitecturas y patrones. Sin embargo, no tenemos tanta experiencia práctica con algunos de ellos.

La arquitectura con la que estamos más familiarizados, ya que hemos diseñado y desarrollado (en asignaturas previas) algunos productos software basados en ella, es la arquitectura REST, así como el patrón Modelo Vista Controlador.

5.1. Arquitectura REST

La arquitectura REST es una arquitectura Cliente-Servidor. El cliente envía peticiones para recuperar o modificar recursos, y el servidor responde con el resultado, que puede ser con los datos que hemos pedido o el estado de la petición.

En la arquitectura REST, una petición está formada por un verbo HTTP que define la operación a realizar. Este verbo puede ser **GET** (recupera información sobre un recurso), **POST** (crea un nuevo recurso), **PUT** (actualiza un recurso) o **DELETE** (elimina un recurso). Las respuestas del servidor contienen códigos de estado donde indica si la petición ha resultado exitosa o no.

5.2. Patrón MVC

El Modelo Vista Controlador (MVC) es el patrón arquitectónico que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos:

- El **Modelo** contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- La **Vista** o interfaz de usuario compone la información que se envía al cliente y los mecanismos de interacción con este.
- El **Controlador**, que actúa como intermediario entre el Modelo y la Vista, gestiona el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

6. Conclusiones

Tras la redacción de este documento y puesta en común del punto de vista de los distintos miembros del equipo, hemos sacado varias conclusiones relacionadas con nuestra experiencia en la arquitectura del software y cómo enfrentarnos al nuevo proyecto que tenemos sobre la mesa.

Aunque no lo sabemos todo sobre las distintas arquitecturas del software, tenemos un conocimiento teórico bastante amplio sobre lo que es la arquitectura del software, lo que es un patrón arquitectónico y lo que es el diseño del software.

Hemos estudiado diferentes tipos de arquitecturas de software y patrones de software, aunque en su mayoría en un marco más teórico. Sin embargo, todos coincidimos en que sabemos desenvolvernos en el uso y la comprensión de las arquitecturas Cliente-Servidor, en concreto en los Servicios REST, que hemos usado en la mayoría de proyectos desarrollados o planificados a lo largo de nuestra vida. También conocemos en profundidad el patrón Modelo Vista Controlador (MVC), que también hemos utilizado en otros proyectos de otras asignaturas.

Una vez puesto en común los conocimientos con los que contamos, hemos concluido con la idea de que, aunque requerirá un previo estudio de la estructura y del framework del proyecto al que nos enfrentamos, seremos capaces de adaptarnos a las necesidades y la arquitectura del nuevo proyecto para poder sacar el producto software adelante, contando con la ayuda de todos los miembros del equipo y el material proporcionado por la asignatura Diseño y Pruebas II.

7. Referencias bibliográficas

-Sección 4:

Wikipedia (Patrones de arquitectura):

https://es.wikipedia.org/wiki/Patrones_de_arquitectura

-Sección 5:

Arquitectura REST. Qué es y para qué sirve:

<https://www.marcpampols.net/es/arquitectura-rest-que-es-para-que-sirve/#:~:text=La%20arquitectura%20REST%20se%20basa,define%20la%20operaci%C3%B3n%20a%20realizar>

Servicio de informática Universidad de Alicante (Modelo Vista Controlador):

<https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>